/***
 * Project Report Template
 * Project 3 (Map Routing), ECE368
 ***/

Name: Shu Hwai Teoh
Login: teoh0

/***
 * Explain your overall approach to the problem and a short
 * general summary of your solution and code.
 ***/

I use Dijkstra's algorithm to find the shortest path between two vertices. Assume the number of the vertex is n, and union contains the start vertex and visited vertices.:

1. Use nx3 matrix to store the vertex number and its x and y coordinates.
2. Build an array that has n slott. Each slott points to an adjacent list.
3. Build an array that has n slott. Each slott points to a node storing the vertex number and its distance from the union.
4. Use Dijkstra's algorithm to find the shortest path between two vertices:
   (1)initialize three arrays that have n spaces:
      a. distance array: use to store the distance from each vertex to the union.
      b. previous array: use to store the previous vertex visited on the shortest path.
      c. visit array: use to record whether a vertex is visited
   (2) Dijkstra's algorithm:
      a. Find an unvisited vertex (A) the that is the nearest from the start vertex or union and calculate the distance (D) between them.
      b. Use A as the start vertex to update the content of distance array and previous array.
      c. Repeat step a and b until the nearest vertex is destination vertex.

/***
 * Known bugs / limitations of your program / assumptions made.
 ***/

Since the largest distance between two vertices is sqrt(10000^2+10000^2)*(n-1)<1500000000, I define the INF as 1500000000. The libraries used includes <stdio.h>, <stdlib.h>, and <math.h>.

/***
 * List whatever help (if any) that you received.
 ***/

Wikipedia and classmates discussion.

/***
 * Describe any serious problems you encountered.
 ***/

I use adjacent matrix to implement Dijkstra's algorithm first and found that there was not enough contiguous memory spaces to store if the number of the vertices larger than 10000. Therefore, I rewrite my program with adjacent list.

/***
 * List any other comments/feedback here (e.g., whether you
 * enjoyed doing the exercise, it was too easy/tough, etc.).
 ***/

I think this project is useful to let us learn something about GPS implementation. Although it is not easy, it is easier to debug than project 2. My program needs to be compiled with the command: gcc -Werror -Wall -O3 main.c -o main -lm